WHAT IS CLAIMED IS:

1. A method of processing messages, comprising:

issuing an input operation from an application to a socket; and

configuring the socket, with the input operation, to recognize a format of a message to be received from a sender, whereby the socket is configured to receive the message without invoking the application until the message is completely received.

2. The method of claim 1, wherein the message is a client-server message.

3. The method of claim 1, wherein issuing the input operation comprises providing to the socket a record definition specifying the format of the message.

4. The method of claim 2, further comprising, issuing an output operation from the socket to the application indicating that the request has been completely received.

5. The method of claim 2, further comprising:

placing, by the socket, the request on a queue when the request is completely received; and

dequeuing, by the application, the request from the queue.

6. The method of claim 2, further comprising utilizing, by the socket, the record definition to determine one of (i) a length of the message and (ii) a terminating character of the message.

7. The method of claim 2, further comprising, interpreting, by the socket, a record header of the message, wherein the record header comprises length information specifying a length of the message.

8. The method of claim 2, further comprising, determining, by the socket, whether the message contains a terminating character.

9. The method of claim 2, wherein only a single input operation from the application layer to the sockets layer is needed before processing the message by the application.

10. The method of claim 2, wherein the message is formatted as a streaming protocol.

11. A computer, comprising:

a network facility configured for transmission of information to and from at least one other computer;

a processor;

a memory containing at least one application and a sockets application programming interface (API); wherein the sockets API, when executed by the processor, is configured to recognize a format of a message received from the at least one other computer, whereby the sockets API is configured to receive the message without invoking the application until the message is completely received.

12. The computer of claim 11, wherein the computer and the at least one computer comprise a client-server environment.

13. The computer of claim 11, wherein the memory further contains a record definition, which when provided to the socket by the application, configures the socket to recognize the format of the message.

14. The computer of claim 11, wherein the format is one of a length field format and a terminating character format, wherein in the length field format the message is configured with a length field indicating a length of the message and in the terminating character format the message is configured with at least one terminating character indicating an end of the message.

15. A data structure contained in a memory of a sockets based system, wherein the data structure, when provided to a socket, configures the socket to recognize a format of a message to be received from a sender, whereby the socket is configured to receive the message without invoking an application to service the message until the message is completely received, the data structure comprising one of:

length field information, if the message is configured with a length field specifying a length of the message; and

terminating character information, if the message is configured with at least one terminating character specifying an end of the message.

16. The data structure of claim 15, wherein the message is a client-server message.

17. The data structure of claim 15, wherein the length field information specifies at least a length of the length field and wherein the terminating character information specifies at least a location of the at least one terminating character in the message.

18. The data structure of claim 15, wherein the length field information comprises:

a size of a message record header containing the length field;

a size of the length field;

a length field descriptor indication whether the length of the message specified by the length field includes the record header; and

an offset within the record header at which the length field begins.

19. The data structure of claim 15, wherein terminating character information comprises:

a pointer to information denoting the end of the message; and

a length of the information denoting the end of the message.

20. A computer-readable medium, containing a program which, when executed, performs an operation of processing client-server messages, the operation comprising:

in response to an input operation from an application to a socket, configuring the socket to recognize a format of a message to be received from a sender, whereby the socket is configured to receive the message without invoking the application until the message is completely received.

21. The computer-readable medium of claim 20, wherein the message is a client-server message.

22. The computer-readable medium of claim 20, wherein the input operation provides to the socket a record definition specifying the format of the message.

23. The computer-readable medium of claim 22, further comprising, issuing an output operation from the socket to the application indicating that the request has been completely received.

24. The computer-readable medium of claim 22, further comprising:

placing, by the socket, the request on a queue when the request is completely received, wherein the request may be subsequently dequeued by the application.

25. The computer-readable medium of claim 22, utilizing, by the socket, the record definition to determine one of (i) a length of the message and (ii) a terminating character of the message.

26. The computer-readable medium of claim 22, further comprising, interpreting, by the socket, a record header of the message, wherein the record header comprises length information specifying a length of the message.

27. The computer-readable medium of claim 22, further comprising, determining, by the socket, whether the message contains a terminating character.

28. The computer-readable medium of claim 22, wherein only a single input operation from the application layer to the sockets layer is needed before processing the message by the application.

29. The computer-readable medium of claim 22, wherein the message is formatted as a streaming protocol.

30. A method of processing messages, comprising:

receiving, at a sockets layer of a computer, data from a remote source via a network connection prior to allocating a buffer to contain the data; and subsequently allocating the buffer to contain the data.

31. The method of claim 30, wherein the messages are client-server messages.

32. The method of claim 30, wherein the data is received over a sockets streaming protocol.

33. The method of claim 30, wherein allocating the buffer comprises sizing the buffer according to a size of the data.

34. The method of claim 30, wherein the allocating is performed in response to a buffer request from the sockets layer.

35. The method of claim 30, wherein the network connection is a Transfer Control Protocol/Internet Protocol (TCP/IP) connection.

36. The method of claim 30, wherein allocating the buffer comprises:
processing a buffer request from a sockets layer after receiving the data; and
providing the buffer to the sockets layer.

37. The method of claim 36, wherein the buffer request specifies a size of the buffer equal to a size of the data.

38. A computer readable medium containing a program which, when executed by a computer, performs operations for processing messages, the operations comprising:
processing an input operation issued from a sockets server application to a sockets layer of the computer, wherein the input operation is configured with a buffer mode parameter indicating to the sockets layer a buffer acquisition method for acquiring a buffer for containing data received from a remote source via a network connection.

39. The computer readable medium of claim 38 wherein the messages are client-server messages.

40. The computer readable medium of claim 38, wherein the data is received over a sockets streaming protocol.

41. The computer readable medium of claim 38, wherein the input operation is further configured with a record definition specifying to the sockets layer a format of the data.

42. The computer readable medium of claim 38, further comprising:
receiving the data from the remote source via the network connection; and
subsequently
allocating the buffer.

43. The computer readable medium of claim 42, wherein the allocation is performed by one of the sockets server application and the sockets layer.

44. The computer readable medium of claim 39, wherein the buffer is allocated from one of:
storage owned by the sockets server application; and
system-supplied storage not owned by the sockets server application.

45. The computer readable medium of claim 39, wherein allocating the buffer comprises sizing the buffer according to a size of the data.

46. The computer readable medium of claim 39, wherein allocating the buffer comprises calling back to the sockets server application with an instruction to allocate the buffer.

47. The computer readable medium of claim 39, wherein the allocating is performed in response to a buffer request made by the sockets layer.

48. The computer readable medium of claim 38, further comprising:
receiving the data from the remote source via the network connection; and
if the buffer is large enough to contain the data, copying the data into a previously allocated buffer provided to the sockets layer with the input operation; and
if the previously allocated buffer is not large enough to contain the data, requesting a larger buffer sufficient to contain the data in accordance with the buffer acquisition method.

49. A system in a distributed environment, comprising:

a network interface configured to support a network connection with at least one other computer in the distributed environment;

a memory comprising a sockets server application, a socket in communication with the sockets server application and a protocol stack in communication with the socket, wherein the protocol stack is configured to transport messages between the network interface and the socket;

a processor configured to perform operations for processing messages, the operations comprising:

processing an input operation issued from the sockets server application to the socket, wherein the input operation is configured with a buffer mode parameter indicating to the socket a buffer acquisition method for acquiring a buffer for containing data received from the at least one other computer.

50. The system of claim 49, wherein the messages are client-server messages.

51. The system of claim 49, wherein the protocol stack is configured for a sockets streaming protocol.

52. The system of claim 49, wherein the memory comprises record definition specifying to the socket a format of the data.

53. The system of claim 49, wherein the operations further comprise:

receiving the data; and subsequently

allocating the buffer.

54. The system of claim 53, wherein the allocation is performed by one of the sockets server application and the socket.

55. The system of claim 53, further comprising application-supplied storage owned by the sockets server application and system-supplied storage not owned by the sockets server application and wherein allocating the buffer comprises one of:

allocating the buffer from application-supplied storage; and

allocating the buffer from system-supplied storage.

56. The system of claim 53, wherein allocating the buffer comprises sizing the buffer according to a size of the data.

57. The system of claim 53, wherein allocating the buffer comprises calling back to the sockets server application with an instruction to allocate the buffer.

58. The system of claim 53, wherein the allocating is performed in response to a buffer request made by the socket.

59. A method of processing messages in a computer, comprising:
providing a system-supplied buffer to a sockets server application;
reading data into the system-supplied buffer; and
sending the data from the system-supplied buffer to another computer via a network.

60. The method of claim 59, wherein the messages are client-server messages.

61. The method of claim 59, wherein the data is sent over a sockets streaming protocol.

62. The method of claim 59, wherein the system-supplied buffer is provided to the sockets server application from a socket of the computer and wherein sending comprises:
returning the system-supplied buffer to the socket of the computer via an application request; and
detaching the system-supplied buffer from the application request to allow the sockets server application to continue processing while sending the data.

63. The method of claim 59, wherein sending is performed without first copying the data into another buffer.

64. The method of claim 59, wherein the reading is performed by the sockets server application.

65. The method of claim 59, further comprising, prior to providing the system-supplied buffer to the sockets server application:

receiving, by a socket, other data from the another computer via the network; and

allocating the system-supplied buffer to contain the other data.

66. The method of claim 59, wherein providing the system-supplied buffer to the sockets server application comprises acquiring, by a socket, the system-supplied buffer from memory space not owned by the sockets server application.

67. The method of claim 59, wherein the system-supplied buffer is provided to the sockets server application by a socket in response to a buffer acquisition function call from the sockets server application.

68. The method of claim 59, wherein the system-supplied buffer is provided to the sockets server application by a socket after the sockets server application requests client data received on a client connection with the another computer.

69. The method of claim 68, wherein providing the system-supplied buffer comprises allocating the system-supplied buffer according to a size of the client data.

70. A computer readable medium containing a sockets-based communications program which, when executed by a computer, performs operations for processing messages, the operations comprising:

providing a system-supplied buffer to a sockets server application;

receiving the system-supplied buffer from the sockets server application, wherein the system-supplied buffer contains data read into the system-supplied buffer by the sockets server application; and

sending the data from the system-supplied buffer to another computer via a network.

71. The computer readable medium of claim 70, wherein the messages are client-server messages.

72. The computer readable medium of claim 70, wherein the providing is performed by a socket of the computer and wherein receiving comprises receiving the system-supplied buffer by the socket on an application request and wherein sending comprises detaching the system-supplied buffer from the application request to allow the sockets server application to continue processing while the data is sent.

73. The computer readable medium of claim 70, wherein sending is performed without first copying the data into another buffer.

74. The computer readable medium of claim 70, wherein the reading is performed by the sockets server application.

75. The computer readable medium of claim 70, further comprising, prior to providing the system-supplied buffer to the sockets server application:

receiving, by a socket, other data from the another computer via the network; and

allocating the system-supplied buffer to contain the other data.

76. The computer readable medium of claim 70, wherein providing the system-supplied buffer to the sockets server application comprises acquiring, by a socket, the system-supplied buffer from memory space not owned by the sockets server application.

77. The computer readable medium of claim 70, wherein the system-supplied buffer is provided to the sockets server application by a socket in response to a buffer acquisition function call from the sockets server application.

78. The computer readable medium of claim 70, wherein the system-supplied buffer is provided to the sockets server application by a socket configured by a receive operation issued from the sockets server application and wherein the system-supplied buffer contains client data from the another computer.

79. The computer readable medium of claim 78, wherein providing the system-supplied buffer comprises allocating the system-supplied buffer according to a size of the client data.

80. The computer readable medium of claim 78, wherein the receive operation is configured with a buffer mode parameter indicating to the socket a buffer acquisition method for acquiring system-supplied buffer.

81. The computer readable medium of claim 80, wherein the receive operation is further configured with a record definition specifying to the socket a format of the client data.

82. A computer in a distributed environment, comprising:

a network interface configured to support a network connection with at least one other computer in the distributed environment;

a memory containing contents comprising:

an operating system;

a sockets server application;

a sockets-based communication facility;

system-owned memory space from which to allocate system-supplied buffers; and

application-owned memory space owned by the sockets server application; and

a processor configured by at least a portion of the contents to perform operations for processing client-server messages, the operations comprising:

providing a system-supplied buffer to the sockets server application for use in sending data to the at least one other computer.

83. The computer of claim 82, wherein the distributed environment is a client-server environment.

84. The computer of claim 82, wherein the protocol stack is configured for a sockets streaming protocol.

85. The computer of claim 82, wherein the processor is configured to send the data without first copying the data into another buffer.

86. The computer of claim 82, wherein providing the system-supplied buffer to the sockets server application comprises acquiring, by the socket, the system-supplied buffer from the system-owned memory space.

87. The computer of claim 82, wherein the operations performed by the processor further comprise:

reading data into the system-supplied buffer;

returning the system-supplied buffer to the socket; and

sending the data from the system-supplied buffer to the at least one other computer.

88. The computer of claim 87, wherein the system-supplied buffer is returned to the socket on a send operation and wherein sending comprises detaching the system-supplied buffer from the send operation to allow the sockets server application to continue processing while the data is sent.

89. The computer of claim 82, wherein the processor is configured to provide the system-supplied buffer to the sockets server application by the socket in response to a buffer acquisition function call from the sockets server application.

90. The computer of claim 82, wherein the socket is configured by a receive operation issued from the sockets server application and configured with a buffer mode parameter indicating to the socket a buffer acquisition method for acquiring system-supplied buffer and wherein the system-supplied buffer contains client data from the at least one other computer.

91. The computer of claim 90, wherein providing the system-supplied buffer comprises allocating the system-supplied buffer according to a size of the client data.

92. The computer of claim 90, wherein the receive operation is further configured with a record definition specifying to the socket a format of the client data.

93. A method of processing messages, comprising issuing a continuous mode input operation from an application to a socket wherein the continuous mode input operation is selected from at least one of:

a single continuous mode accept operation configuring a listening socket to handle a plurality of incoming client connections; and

a single continuous mode receive operation configuring a client socket to handle a plurality of client requests.

94. The method of claim 93, wherein the messages are client-server messages.

95. The method of claim 93, further comprising, configuring the client socket, with the single continuous mode receive operation, to recognize a format of each of the plurality of client requests, whereby the client socket is configured to receive the client requests without invoking the application until the request is completely received.

96. The method of claim 93, wherein the continuous mode input operations are issued from a main thread of the application.

97. The method of claim 93, wherein issuing the single continuous mode receive operation comprises:

placing a single pending receive data structure on a pending queue;

for each completed client request, copying contents of the pending receive data structure to a completed receive data structure queued on a receive completion queue.

98. The method of claim 93, wherein issuing the single continuous mode accept operation comprises:

placing a single pending accept data structure on a pending queue;

for each of the plurality of incoming client connections, copying contents of the single pending accept data structure to a completed accept data structure queued on a accept completion queue, wherein the single pending accept data structure remains on the pending queue.

99. The method of claim 98, wherein issuing the single continuous mode receive operation comprises:

placing a single pending receive data structure on a pending queue;

for each completed client request, copying contents of the pending receive data structure to a completed receive data structure queued on a receive completion queue.

100. The method of claim 93, further comprising, for each completed client request, acquiring a buffer from system supply memory to contain the completed client request.

101. The method of claim 100, wherein allocating the buffer comprises sizing the buffer according to a size of the completed client request.

102. A computer readable medium containing a sockets-based program comprising at least one of a continuous mode accept application programming interface and a continuous mode receive application programming interface, wherein the sockets-based program, when executed, performs operations for processing messages, the operations comprising at least one of:

configuring a listening socket to handle a plurality of incoming client connections as a result of issuing a single continuous mode accept operation from an application; and

configuring a client socket to handle a plurality of client requests as a result of a single continuous mode receive operation issued by the application.

103. The computer readable medium of claim 102, wherein the messages are client-server messages.

104. The computer readable medium of claim 102, further comprising, configuring the client socket, with the single continuous mode receive operation, to recognize a format of each of the plurality of client requests, whereby the client socket is configured to handle receiving the client requests without invoking the application until the message is completely received.

105. The computer readable medium of claim 102, wherein the continuous mode accept operation and the continuous mode receive operation operations are issued from a main thread of the application.

106. The computer readable medium of claim 102, further comprising, when the single continuous mode receive operation is issued:

placing a single pending receive data structure on a pending queue;

for each completed client request, copying contents of the pending receive data structure to a completed receive data structure queued on a receive completion queue.

107. The computer readable medium of claim 102, further comprising, when the single continuous mode accept operation is issued:

placing a single pending accept data structure on a pending queue;

for each of the plurality of incoming client connections, copying contents of the single pending accept data structure to a completed accept data structure queued on a accept completion queue, wherein the single pending accept data structure remains on the pending queue.

108. The computer readable medium of claim 107, further comprising, when the single continuous mode receive operation is issued:

placing a single pending receive data structure on a pending queue;

for each completed client request, copying contents of the pending receive data structure to a completed receive data structure queued on a receive completion queue.

109. The computer readable medium of claim 102, further comprising, for each completed client request, acquiring a buffer from system owned memory space to contain the completed client request.

110. The computer readable medium of claim 109, wherein allocating the buffer comprises sizing the buffer according to a size of the completed client request.

111. A system in a distributed computer environment, comprising:

a network facility configured to support a network connection with a remote computer;

a memory containing content comprising an application and a plurality of sockets application programming interfaces (APIs), wherein the sockets APIs comprise at least one of a continuous mode accept operation and a continuous mode receive operation;

a processor which, when executing the contents, is configured to perform operations comprising at least one of:

issuing a single continuous mode accept operation to configure a listening socket to receive a plurality of incoming client connections; and

issuing a single continuous mode receive operation to configure a client socket to receive a plurality of client requests.

112. The system of claim 111, wherein the distributed computer environment is a client-server environment.

113. The system of claim 111, wherein the content of the memory further comprises a system owned memory space and wherein the operations further comprise:

for each completed client request, acquiring a buffer from the system owned memory space to contain the completed client request.

114. The system of claim 111, wherein the content of the memory further comprises a system owned memory space and wherein the operations further comprise:

for each completed client request, acquiring a buffer from the system owned memory space to contain the completed client request, wherein the buffer is sized according to a size of the completed client request.

115. The system of claim 111, wherein the content of the memory further comprises a pending queue on which a single pending accept data structure is queued as a result of the continuous mode accept operation.

116. The system of claim 115 wherein the content of the memory further comprises an accept completion queue to which contents of the pending accept data structure are copied upon receiving a client connection on the listening socket and wherein the pending accept data structure remains on the pending queue.

117. The system of claim 111, wherein the content of the memory further comprises a pending queue on which a single pending receive data structure is queued as a result of the continuous mode receive operation.

118. The system of claim 117, wherein the content of the memory further comprises a receive completion queue to which contents of the pending receive data structure are copied upon receiving a completed client request on the client socket and wherein the pending receive data structure remains on the pending queue.